Lab 09 Template – Ethan Roepke

Some questions require multiple parts to be answered, be sure to discuss them in full.

1) **Screenshot of the output from openssl ec -text -in <netid>_private_key.pem |less (5 points)**

```
read EC key
writing EC key
Private-Key: (384 bit)
priv:
    dc:f7:2b:e5:8c:0c:8d:3e:be:42:12:c6:cc:75:6b:
    da:7a:d7:84:67:d4:fa:44:8b:54:5c:3f:c1:29:3f:
    f1:05:ea:92:60:b3:a0:d4:81:f6:5a:77:00:4a:17:
    33:16:35
pub:
    04:17:0c:b4:1c:3c:c1:70:b0:6b:c2:10:10:61:2c:
    1b:0c:fa:63:c0:52:df:7d:4f:38:c1:90:7d:ab:06:
    a9:8b:62:43:79:f9:58:ab:5b:bb:3c:7b:10:e5:be:
    c4:3d:b8:2c:e8:b7:b1:ba:a3:df:53:fc:09:d6:76:
    a5:c5:9a:0f:43:8b:e3:57:f1:04:e3:8e:10:33:d2:
    d4:fa:42:32:13:e5:ee:73:8f:92:3c:69:e4:92:61:
    56:d6:61:ac:02:f3:85
ASN1 OID: secp384r1
NIST CURVE: P-384
-----BEGIN EC PRIVATE KEY-----
MIGkAgEBBDDc9yvljAyNPr5CEsbMdWvaeteEZ9T6RItUXD/BKT/xBeqSYLOg1IH2
WncAShczFjWgBwYFK4EEACKhZANiAAQXDLQcPMFwsGvCEBBhLBsM+mPAUt99TzjB
kH2rBqmLYkN5+VirW7s8exDlvsQ9uCzot7G6o99T/AnWdqXFmg9Di+NX8QTjjhAz
0tT6QjIT5e5zj5I8aeSSYVbWYawC84U=
-----END EC PRIVATE KEY-----
(END)
```

2) **What is the difference between the two curve identifiers? (5 points)**
ASN1 OID follows the naming used by the SECG and includes information about the curves parameters and structure. NIST CURVE has standardized curves and each of these curves is given a different name. The main difference is ANS1 OID is used at the technical protocol level while NIST CURVE is used is security guidelines.

3) **What does the CONNECTED(00000003) line indicate about the steps that occurred before establishing the SSL/TLS handshake? (5 points)**
The CONNECTED(00000003) line lets us know that we connected successfully to server as a client and established a TCP connection before establishing SSL/TLS handshake.

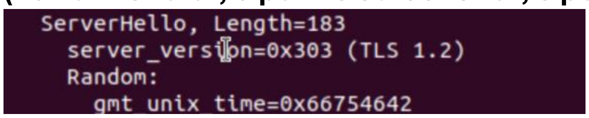
4) **Take a screenshot of the client key_share information(5 points)**

```
    psk_dhe_ke (1)
extension_type=key_share(51), length=103
    NamedGroup: secp384r1 (P-384) (24)
    key_exchange:  (len=97): 045739631DFD054B78DD0873A0E049846C3306CB034FDBC2C60F6324232C306FD9C13F523B0DAE0F2C007EDCC9C808F952E5C87B1EDC6EB9C543BF029E14B7308AD3737281D3581469082F2337FB9BC523A9769AC16EA8A9C0E435FB6B3F8A5E16
```

5) **What information do you think the client key_share fields are providing to the server? How might the server use this information during the TLS 1.3 handshake to establish a secure connection? Does this contribute to perfect forward secrecy? Why or why not?(10 points total, 2.5 points each question)**

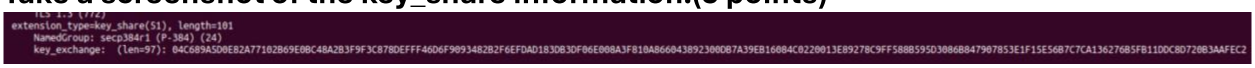
The client key_share field is providing information about the clients public key and elliptical curve(secp384r1) to use for key exchange. The server may use this information to generate its own public key to exchange to the client. It would also use its private key and public key from client so both can encrypt the message or whatever.

6) **Take a screenshot of the supported version extension for the server. What version(s) of TLS does the server actually support?**
(10 Points total, 5 points screenshot, 5 points question)

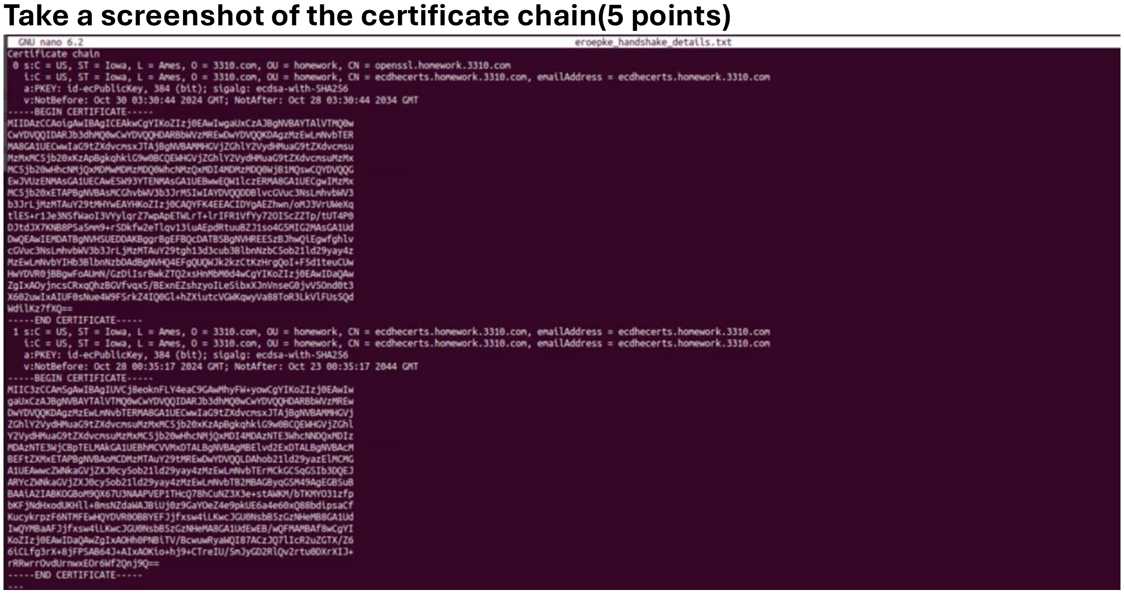


The versions of TLS that the server actually support is TLS 1.3 from the supported_version given a few lines under the image above.

7) **Take a screenshot of the key_share information.(5 points)**



8) **Take a screenshot of the certificate chain(5 points)**



9) **Look at the subject (s) and the issuer (i) for each of the two certificates. Which number is the certificate for the Certificate Authority (CA)? Why? (10 points total, 5 points for each question)**
0 lists itself as issued by the entity at next depth level. 0 relies on 1 as its issuer. 1 is for a client that uses its self signed certificates. They both should be same for subject and issuer.

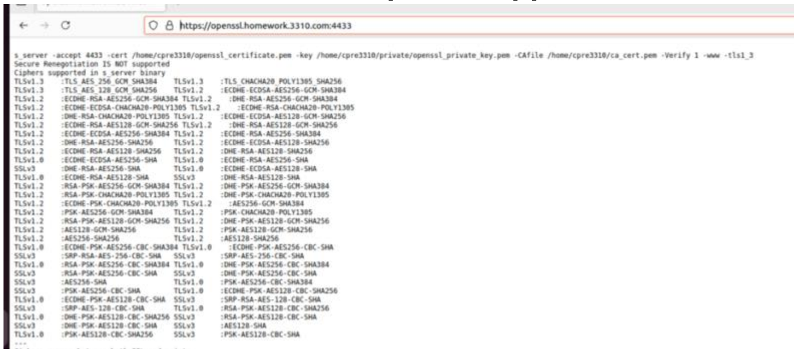
10) **What happens if the server's certificate validation fails at any depth?(5 points)**
If servers certificate validation fails at any depth, the handshake that we were running would be terminated. The client will lose connection the servers certificates to prevent against attacks.

11) **Expand the acronym for each piece of the cipher suite and explain the purpose of each.(10 points total, 2 points each piece)**
TLS_AES_256_GCM_SHA384
TLS: Transport Layer Security
    Provides a secure communication over a network
AES: Advanced Encryption Standard
    Symmetric encryption algorithm that is used to secure data
256: 256 bits is key length
    Large size for bit key to make it difficult for attackers to use brute force
GCM: Galois and Counter mode
    Encryption mode that provides both encryption and authentication
SHA384: Security Hash Algorithm(384 bits)
    cryptographic hash function that generates a 384-bit hash value, used for message authentication

12) **What does "getting new key material" mean? (5 points)**
"getting new key material" means for every session, new keys for encryption are randomly generated. This is a security so if one session does get compromised, no other sessions can get compromised because it is unique to itself.

13) **Why would creating pkcs12 files be a bad idea generally speaking?(5 points)**
Creating pkcs12 files is generally a bad idea because we are saving the private key and the certificate in a single file. This means if an attacker can compromise that one file, they gather 2 sensitive parts.

14) **Take a screenshot of the ciphers supported in s_server binary. (5 points)**



15) **Find and record in your lab report at least one that should no longer be used in the real world. Support your reason why it shouldn't be used. (10 points total)**

One cipher supported in s_server binary from the list is SSLv3. This should no longer be used in the real world because this is an outdated protocol and vulnerable to attacks. SSLv3 is vulnerable to an attack called POODLE, which downgrades a user's browser to SSL 3 then exploits the vulnerability to decrypt/extract information from encrypted transactions